Научная статья
УДК 004.832; 004.853
doi:10.37614/2949-1215.2025.16.3.004

ПРИМЕНЕНИЕ RAG-ТЕХНОЛОГИИ ДЛЯ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ТЕСТОВ И ПРОВЕРКИ ЗНАНИЙ С ПОДДЕРЖКОЙ ДИАЛОГА НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Алексей Владимирович Шестаков[1]✉, Александр Анатольевич Зуенко[2]
[1, 2]Институт информатики и математического моделирования имени В. А. Путилова Кольского научного центра Российской академии наук, Апатиты, Россия
[1]a.shestakov@ksc.ru✉, https://orcid.org/0000-0002-9052-25791
[2]a.zuenko@ksc.ru, https://orcid.org/0000-0002-7165-6651

Аннотация

Работа посвящена применению RAG-технологий (Retrieval-Augmented Generation) для автоматической генерации тестов и адаптивной проверки знаний с поддержкой диалога на естественном языке. Проведен анализ аналогичных систем поддержки исследовательской деятельности для генерации вопросов. В работе представлена общая архитектура системы, а также детально рассмотрены системы генерации вопросов и адаптивного тестирования, описаны особенности реализации модулей генерации тестовых вопросов и динамического подбора заданий на основе ответов пользователя. Приведены результаты тестирования и сделаны выводы о практической применимости системы.

Ключевые слова:

RAG-технологии, большие языковые модели, интеллектуальный поиск, вопросно-ответная система, обработка естественного языка, онтологии, адаптивное тестирование, научная деятельность

Благодарности:

работа выполнена в рамках научно-исследовательской работы «Методы и информационные технологии мониторинга и управления региональными критическими инфраструктурами Арктической зоны Российской Федерации» (FMEZ-2025-0054).

Для цитирования:

Шестаков А. В., Зуенко А. А., Применение RAG-технологии для автоматической генерации тестов и проверки знаний с поддержкой диалога на естественном языке // Труды Кольского научного центра РАН. Серия: Технические науки. 2025. Т. 16, № 3. С. 56–70. doi:10.37614/2949-1215.2025.16.3.004.

Original article

APPLICATION OF RAG TECHNOLOGY FOR AUTOMATED TEST GENERATION AND KNOWLEDGE ASSESSMENT WITH NATURAL LANGUAGE DIALOGUE SUPPORT

Aleksey V. Shestakov[1]✉, Alexander A. Zuenko[2]
[1, 2]Putilov Institute for Informatics and Mathematical Modeling of the Kola Science Centre of the Russian Academy of Sciences, Apatity, Russia
[1]a.shestakov@ksc.ru✉, https://orcid.org/0000-0002-9052-25791
[2]a.zuenko@ksc.ru, https://orcid.org/0000-0002-7165-6651

Abstract

This study explores the application of Retrieval-Augmented Generation (RAG) technology for the automated generation of tests and adaptive knowledge assessment, supported by natural language dialogue. An analysis of similar research support systems for question generation is conducted. The paper presents the overall system architecture and provides a detailed examination of its core components: the question generation subsystem and the adaptive testing engine. The implementation specifics of the test question generation module and the dynamic task selection mechanism based on user responses are described. The results of testing are presented, followed by conclusions regarding the system's practical applicability.

Keywords:

retrieval-augmented generation (rag), large language models (LLMs), intelligent information retrieval, question-answering systems, natural language processing (NLP), ontologies, adaptive testing, research activity

Acknowledgements:

This work was supported by the research project “Methods and Information Technologies for Monitoring and Managing Regional Critical Infrastructures in the Arctic Zone of the Russian Federation” (Project No. FMEZ-2025-0054).

Труды Кольского научного центра РАН. Серия: Технические науки. 2025. Т. 16, № 3. С. 56–70.
Transactions of the Kola Science Centre of RAS. Series: Engineering Sciences. 2025. Vol. 16, No. 3. P. 56–70.

For citation:

Shestakov A. V., Zuenko A. A. Application of RAG technology for automated test generation and knowledge assessment with natural language dialogue support. *Trudy Kol'skogo nauchnogo centra RAN. Seriya: Tekhnicheskie nauki* [Transactions of the Kola Science Centre of RAS. Series: Engineering Sciences], 2025, Vol. 16, No. 3, pp. 56–70. doi:10.37614/2949-1215.2025.16.3.004.

Введение

Научные сотрудники постоянно имеют дело с поиском и анализом информационных источников. Однако, несмотря на доступность информации, ее объемы быстро растут. Ежедневно выпускается огромное количество публикуемых работ: научных статей, монографий, патентов и технических отчетов. Традиционные методы поиска, такие как поиск по содержимому (полнотекстовый поиск) и по ключевым словам, демонстрируют ограниченность, так как они не учитывают контекст и семантику данных, вследствие чего не способны в значительной мере ускорить и упростить поиск информации. Таким образом, возрастает потребность в контекстно-ориентированных интеллектуальных системах поиска и анализа информации, которые способны понимать семантику информации при интеллектуальном анализе.

Отличительной чертой научных работ является специфика научной информации. При разработке интеллектуальных систем поддержки исследовательской деятельности особое внимание уделяется принципиальной разнородности форм представления информации в научных работах, которая может иметь форму текста или быть сгруппированной в таблицы, описываться математическими формулами, схемами, графиками, изображениями. Кроме этого, научная информация может храниться в разных форматах, таких как PDF, DOCX, XLSX. В процессе разработки интеллектуальной системы поддержки исследовательской деятельности для каждого из форматов необходимо применять свои специальные алгоритмы обработки, чтобы сохранить структуру и смысловые связи текста. При написании научных текстов обычно соблюдаются определенные традиции изложения результатов исследований, тексты характеризуются наличием определенной структуры и стилем изложения материала. Помимо изложенного выше, значение терминов может различаться в зависимости от предметной области, что осложняет разработку интеллектуальных поисковых систем.

К таким интеллектуальным системам поддержки исследовательской деятельности предъявляются строгие требования:

1. Система должна обеспечивать высокую степень релевантности получаемого ответа, учитывая контекст: конкретную предметную область, историю диалога с пользователем и его профиль в системе (студент, аспирант, ведущий научный сотрудник и т. п.).

2. Генерируемые системой ответы должны содержательно соответствовать исходным источникам, а также принятым нормам изложения на русском языке, что подразумевает использование устоявшейся терминологии, точность формулировок и соблюдение логики научного стиля.

3. Так как исследовательская деятельность не ограничивается поиском информации, система должна предоставлять широкий спектр дополнительных возможностей. К ним относится проверка усвоения информации. Для этого предлагается применять адаптивное тестирование пользователей, которое позволяет автоматически составлять тесты на основе загруженных материалов для проверки понимания материала и выявления пробелов в знаниях. Это может быть полезно при решении таких задач, как построение индивидуальной траектории обучения, формирование команды под проект.

4. Система должна обеспечивать поддержку диалога на естественном языке как при формулировке запросов (вопросов), так и при анализе ответов пользователя.

Анализ существующих платформ показывает, что имеющиеся интеллектуальные системы поиска не удовлетворяют всем имеющимся требованиям, особенно в части автоматической генерации тестов для проверки знаний.

В качестве подхода к решению задач интеллектуального поиска и автоматизации проверки знаний можно применять подход на основе больших языковых моделей (БЯМ) [1]. Такие модели способны осуществлять поиск и генерировать ответы на естественном языке, а также могут служить основой для генерации вопросов и проведения диалога. Однако существенным недостатком БЯМ является генерация недостоверных сведений — «галлюцинаций» (т. е. неточных или вводящих в заблуждение

результатов), что является серьезным ограничением при применении БЯМ. Для преодоления этого недостатка предлагается использовать гибридные модели с использованием RAG-технологии (Retrieval Augmented Generation) [2]. Совместное применение этих моделей потенциально способно повысить эффективность исследовательского процесса и уменьшить число рутинных задач. Методы RAG-технологии дополняют ответы БЯМ релевантными фрагментами из базы данных и поддерживают семантический поиск с учетом контекста.

Целью настоящего исследования является разработка системы автоматической генерации тестов и проверки знаний с поддержкой диалога на естественном языке. Система должна осуществлять генерацию наборов вопросов на основе загруженных научных материалов, адаптивное тестирование пользователя на основе сгенерированных тестов. В процессе ответов на вопросы теста система должна обеспечивать автоматическую адаптацию сложности вопросов, анализ ответов пользователя и формирование индивидуальной траектории обучения.

Обзор существующих решений

Интеллектуальные системы поддержки исследовательской деятельности являются быстро развивающейся областью. Далее производится анализ существующих решений для задач семантического поиска и анализа научных текстов.

Международные поисковые системы

Наиболее известными интеллектуальными системами поддержки исследовательской деятельности на международном рынке являются Elicit [3], Semantic Scholar [4], IBM Watson Discovery [5]. Эти системы применяют для работы с научными текстами современные методы обработки естественного языка. Тем не менее при их применении для работы с русскоязычными научными текстами возникают существенные проблемы. Во-первых, рассматриваемые системы имеют низкую эффективность при обработке морфологии русского языка и специфической научной терминологии. Во-вторых, их алгоритмы настроены на структуру и стиль зарубежных публикаций, что приводит к существенным стилистическим ошибкам при анализе русскоязычных публикаций. Также для организаций могут возникать сложности при использовании зарубежных систем из-за требований к конфиденциальности. При этом стоит отметить, что ни одна из перечисленных систем не предлагает полноценных возможностей по автоматической генерации тестов и проведению адаптивного тестирования на основе документов пользователя.

На настоящий момент наиболее перспективным подходом к созданию контекстно-ориентированных систем поиска является применение RAG-технологии. Например, в работе [6] предлагается фреймворк для поиска по научной литературе с применением методов семантической фрагментации и стратегии «abstract-first» для быстрой фильтрации статей. Авторы делают основной акцент на тонкой настройке моделей эмбеддингов (фрагментов текста в векторном представлении) под конкретную предметную область для повышения точности результатов поиска. Другой пример — система LitLLM [7] для автоматизации написания литературных обзоров. Основное преимущество предлагаемой системы заключается в интеграции с внешними API (Semantic Scholar [4], OpenAlex [8]) для поиска публикаций и использовании БЯМ для сортировки результатов и генерации текстов. Однако данная система не поддерживает возможность ведения диалога на естественном языке.

Системы автоматической генерации вопросов

Среди систем автоматической генерации вопросов выделяют несколько подходов. Один из подходов к генерации вопросов представлен в работе [9]. В его основе лежит оптимизированная под задачи генерации вопросов нейросетевая модель, поддерживающая механизм внимания. Для анализа контекста и повышения качества понимания семантических связей авторы работы применяют двунаправленные LSTM-сети (Long short-term memory) [10]. Для формирования вопросов доступно два режима работы: a) с учетом только текущего предложения; b) с привлечением более широкого контекста абзаца.

Для обучения модели авторы работы применяют датасеты SQuAD [11] и MS MARCO [12] на парах «предложение — вопрос» и предобученные эмбеддинги (GloVe) [13] для улучшения качества генерации. Для ситуации, когда модель сталкивается с неизвестными словами, в работе [9] предлагается применять технологию замены токенов UNK (от “unknown” — неизвестный). Таким образом, если система сталкивается с незнакомым модели словом, происходит его замена на термин из исходного текста с наибольшим весом в механизме внимания. Тем не менее при практическом применении система не всегда в полной мере способна обеспечить качество генерации и семантической корректности вопросов, как показано на рис. 1.

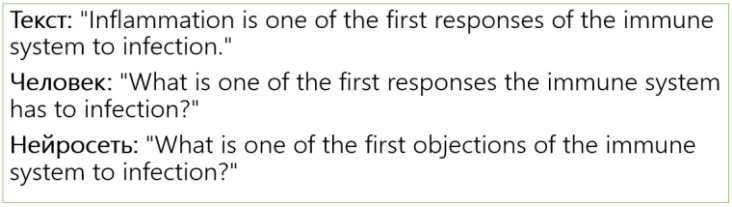

Рис. 1. Пример составления вопроса

В данном примере показано, что система в некоторых случаях генерирует «неправильные» слова, то есть слово “responses” на “objections”.

Таким образом, рассмотренный подход, несмотря на высокую эффективность, требует больших объемов данных и переобучения модели при изменении предметной области, а также иногда система генерирует семантически некорректные слова, а модель с контекстом абзаца не всегда улучшает результаты.

Еще один подход, направленный на применение БЯМ для генерации вопросов, представлен в работе [14]. Авторы работы представляют систему, которая использует дообученные на датасете Turkish-Quiz-Instruct модели GPT-3.5-Turbo и Llama-2 для турецкого языка. Подход ориентирован на генерацию вопросов на основе учебных материалов с предоставлением в качестве ответов набора альтернатив, а также позволяет анализировать краткие ответы на естественном языке. Процесс генерации вопросов включает несколько этапов. На первом этапе происходит сбор и «очистка» учебных материалов, далее выполняется создание промптов для генерации вопросов. После этого происходит использование БЯМ для формирования тестовых вопросов. На последнем этапе система производит оценку качества вопросов с помощью экспертной проверки и метрик ROUGE [15], то есть метрик, которые учитывают, прежде всего, лексическое перекрытие, то есть формальное сходство текстов на уровне слов, а не их смысловую эквивалентность.

В работе предлагается два способа оценки качества вопросов: количественно с применением ROUGE-1, ROUGE-2, ROUGE-L; качественно с использованием экспертных оценок по пятибалльной шкале. По метрикам ROUGE, модель БЯМ GPT-3.5-Turbo обеспечивает более высокую точность генерации, а по результатам экспертной оценки лучше оказывается языковая модель Llama-2-13b-chat-hf.

Таким образом, к преимуществам рассматриваемого подхода относятся: поддержка нескольких форматов вопросов, уменьшение времени на внедрение и требуемой аппаратной производительности по сравнению с подходами, основанными на обучении нейросети за счет применения БЯМ. Использование метрик ROUGE не всегда способно отразить реальное качество вопросов путем оценки поверхностного сравнения сходства без глубокого семантического анализа.

Кратко сформулируем результаты проведенного анализа подходов к генерации тестов с возможностью получения ответов в произвольной форме на естественном языке. Созданные под задачи генерации вопросов нейросетевые модели обеспечивают стабильное качество в пределах своей предметной области, но требуют больших объемов данных и переобучения в случае перехода к другой предметной области. Применение дообученных БЯМ делают систему более адаптивной, однако такие модели зависимы от качества оригинальной модели и «проработки» управляющих промптов.

Существенным недостатком второго подхода является отсутствие интеграции систем генерации вопросов с механизмами тестирования и проверки знаний, а также ориентация на метрики типа ROUGE, которые направленны на оценку поверхностного сходства. Также одной их проблем является

слабая поддержка структурированных данных, представленных в виде таблиц, что критически важно для систем, работающих с научными текстами.

Методология и архитектура системы

Для решения задач интеллектуальной поддержки научной деятельности, которые включают семантический поиск и автоматизированную проверку знаний, предлагается использовать гибридную архитектуру, основанную на применении БЯМ и RAG-технологии.

Методология

Для реализации функций разрабатываемой интеллектуальной системы были выбраны несколько методов: гибридный поиск на основе RAG-технологии; генерация вопросов на основе промпт-инжиниринга; динамическое онтологическое моделирование.

Для реализации базового функционала в рамках настоящего исследования предлагается использовать RAG-технологию, что позволяет применять БЯМ в сочетании с семантическим поиском по базе документов и минимизировать количество «галлюцинаций» у генеративных моделей без необходимости переобучения модели. Применение подобного подхода является важным при работе с научной документацией, поскольку «чистые» языковые модели неспособны предоставить необходимое качество ответов на запросы. Используемый подход способен реализовать генерацию ответов на естественном языке, которые соответствуют стилю речи научного русского языка.

Перейдем к рассмотрению процесса генерации вопросов определенных типов и сложности на основе контекста. БЯМ функционирует с использованием специализированных промптов (специализированных шаблонов для управления поведением языковой модели). Такой подход способен обеспечивать генерацию вопросов приемлемого качества с учетом загруженного контекста. В настоящей статье предлагается алгоритм, который динамически регулирует последовательность и сложность вопросов, которые будут задаваться пользователю. Таким образом, применяемые методы дают системе возможность анализировать ответ пользователя, принимать синонимичные формулировки, а не только сравнивать ответ с заранее заданным строгим шаблоном.

Система не требует переобучения базовой языковой модели, так как и оценка качества ответов, и генерация тестов реализуются преимущественно на основе применения RAG-архитектуры и методов промпт-инжиниринга.

Архитектура системы

Архитектура разработанной системы представлена на рис. 2. Детальное описание компонентов системы и их взаимодействия приведено в работе [16]. Краткое описание основных модулей системы представлено ниже.

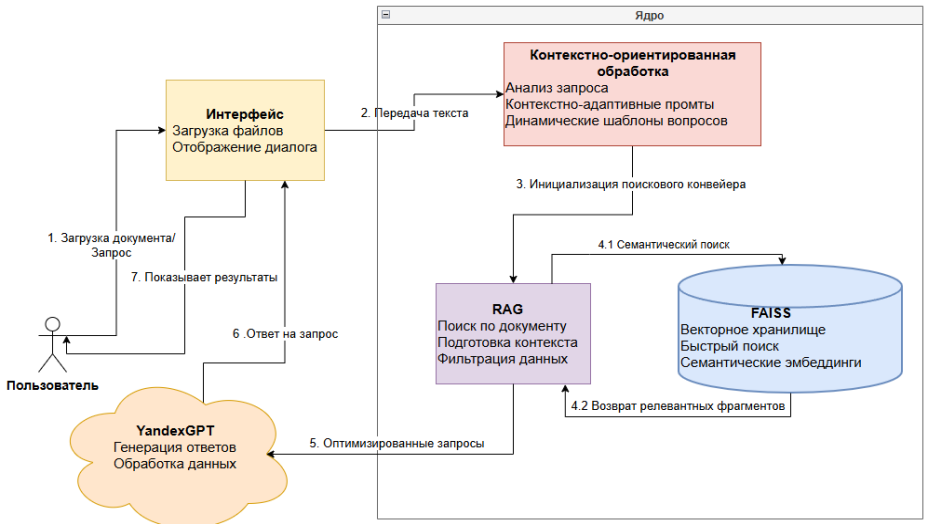

Рис. 2. Архитектура интеллектуальной системы поддержки исследовательской деятельности

Модуль загрузки и обработки документов

В модуле загрузки и обработки документов происходит решение задач, связанных с обработкой различных типов данных и форматов их представления, что влияет на качество всего последующего анализа. Модуль реализует следующий функционал.

Поддержка форматов. Для реализации эффективной обработки и парсинга документов применяются специальные библиотеки и авторские методы. Система поддерживает работу с наиболее распространенными форматами (PDF, DOCX и XLSX), сохраняя структуру и семантические связи исходного текста при обработке документов.

Семантическое фрагментация (чанкирование). При работе с большими объемами частой проблемой являются жесткие ограничения БЯМ по количеству токенов на размер контекста. Стратегия адаптивного семантического чанкирования (деление информации большого объема на небольшие фрагменты) позволяет обойти проблемы, связанные с размером контекста. В отличие от обычного разбиения текста на символы, в процессе фрагментации происходит выделение в тексте логических блоков. В процессе исследований выявлено, что для наиболее эффективного сохранения смысловой целостности информации размер фрагментов должен находиться в диапазоне от 300 до 2000 токенов.

Обработка таблиц. Одним из наиболее часто встречающихся видов представления результатов исследований в научных работах являются таблицы. Для работы с такими структурированными данными авторами реализован механизм обработки таблиц. Система идентифицируют структуру таблицы (заголовки, типы данных, связи между ячейками), а затем производит процесс векторизации, учитывая структурно-семантические связи внутри таблицы. Таким образом, система способна проводить точный семантический поиск по данным внутри таблиц.

Векторное хранилище и RAG-ядро

RAG-технология и векторное хранилище отвечают за семантический поиск и обеспечение приемлемого качества результата.

Для хранения данных используется векторная база данных (ВБД) FAISS (Facebook AI Similarity Search) [17]. После обработки данных на предыдущем этапе получившиеся фрагменты преобразуются в числовые векторные представления (эмбеддинги), которые затем индексируются и помещаются в ВБД. Такой подход позволяет осуществлять поиск фрагментов исходного документа, семантически наиболее схожих с запросом пользователя, в том числе в случаях неполного лексического совпадения.

RAG-технология является центральным звеном при генерации достоверных ответов. В процессе работы технологии можно выделить несколько ключевых шагов: 1) получение запроса пользователя и преобразование в векторное представление; 2) семантический поиск в ВБД наиболее релевантных фрагментов текста (чанков); 3) объединение найденных фрагментов с пользовательским запросом и подходящим к данному типу запроса промптом, а также подача на вход БЯМ; 4) генерация БЯМ связанного ответа на основе представленного контекста. Так как языковая модель опирается исключительно на заданный источник, а не на свои «внутренние знания», этот подход значительно повышает качество ответа за счет снижения «галлюцинаций» модели.

Модуль интеграции с БЯМ (YandexGPT/GigaChat)

В рамках предложенной архитектуры для обеспечения возможности быстрого и простого способа переключения между различными языковыми моделями, как локальными, так и облачными, например российскими YandexGPT [18] и GigaChat [19], используется фреймворк LangChain [20]. Такой подход обладает следующими достоинствами. Во-первых, обеспечивается унифицированный интерфейс для взаимодействия с различными языковыми моделями. Во-вторых, за счет применения методов *промпт-инжениринга* появляется возможность управлять поведением БЯМ, в частности поддерживать стиль ответов, их структуру. Промпт представляет собой специальные четкие инструкции для языковой модели, которые предписывают, как БЯМ должна вести себя при генерации ответов. Наконец, поддерживается возможность хранения истории диалога и учета его при генерации ответов, что дает возможность пользователю задавать уточняющие вопросы, а системе — поддерживать диалог на естественном языке, основываясь на предыдущем взаимодействии.

Блок контекстно-ориентированной обработки информации

Центральным элементом всей разрабатываемой системы является блок контекстно-ориентированной обработки информации (взаимодействия с пользователем). Основная задача блока — классификация запроса пользователя относительно контекста загруженных документов, контекста диалога и профиля пользователя, включая направление запроса в поисковый или тестовый модуль. Блок включает следующие элементы.

Классификатор типа запроса. Выполняет анализ и классификацию входящего запроса пользователя на основании специальных логических правил, определяет его назначение: является ли запросом на поиск информации, командой генерации теста, командой для начала тестирования, ответом на тестовое задание. В случае типовых поисковых запросов система может предоставить заготовленный ответ из базы знаний без применения БЯМ.

Модуль поиска, основанный на RAG-технологии. Осуществляет управление контекстом диалога, которое включает как анализ предыдущей последовательности вопросов-ответов, так и метаинформацию: знания предметной области, контекст документа, профиль и уровень подготовки пользователя. Получившийся из перечисленных элементов расширенный контекст (итоговый промпт) передается на обработку БЯМ.

Модуль адаптивного тестирования. Активируется для типовых запросов пользователя, связанных с операциями по адаптивному тестированию. Его работа подробно описана в следующем подразделе.

Таким образом, блок контекстно-ориентированного поиска является «мозгом» разрабатываемой системы и обеспечивает ведение осмысленного, персонализированного диалога в рамках конкретной предметной области.

Предлагаемый метод генерации вопросов и проверки качества ответов

Генерация тестов

Процесс генерации тестов можно представить в виде UML-диаграммы последовательности на рис. 3.

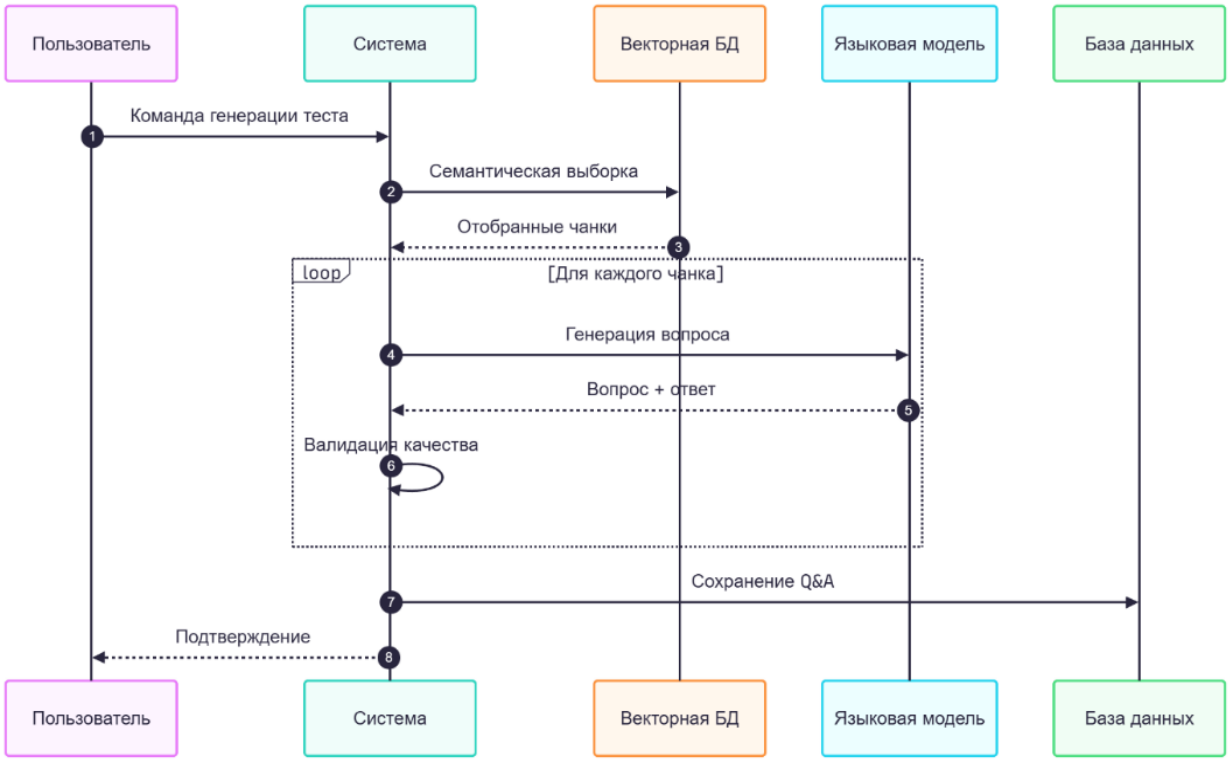

Рис. 3. UML-диаграмма последовательности: генерации тестовой базы

Инициализация. Пользователь дает команду «генерация теста» (или синонимичную команду), после чего система запускает процесс генерации тестовых заданий в соответствии с указанными параметрами (количество вопросов, уровень сложности, тематическая направленность). Происходит обращение к ВБД и семантическая выборка из загруженных материалов фрагментов текста, которые затрагивают разные темы. Алгоритм семантической выборки включает:

1) выделение тем в базе документа, выбранного как контекст для теста;

2) отбор фрагментов таким образом, чтобы обеспечить пропорциональное покрытие тематических разделов;

3) группирование вопросов по сложности;

4) фильтрация фрагментов: исключение вводных фраз и общих рассуждений, а также проверка на наличие релевантных определений и формулировок и т. д.

Критерии отбора семантических фрагментов для генерации по ним вопросов включают: 1) семантическое сходство с ключевой темой (определяется через косинусное сходство между векторными представлениями и должно превышать порог 0,7); 2) степень разнообразия вопросов; 3) семантическая независимость смысловых блоков (например, абзацев).

Создание пары «Вопрос — ответ» (Q&A). Система генерирует заданное число вопросов разных типов и сложности на основе отобранных семантических фрагментов и специализированных промптов. В итоге получается структурированный список «тема — вопрос — эталонный ответ», который сохраняется в БД. Стоит отметить, что при генерации эталонных ответов система основывается исключительно на загруженном контексте.

Процесс формирования вопросов осуществляется с использованием специализированных промптов, которые обеспечивают соответствие заданному стилю и типу вопроса, а также требованиям к сложности вопроса. Пример базового промпта для генерации вопросов показан ниже.

Листинг 1. Пример базового промпта для генерации вопросов

```
Сгенерируй {N} вопросов по следующему контексту, соблюдая требования:
1. **Структура**: раздели вопросы по темам, в каждой теме не менее 3 вопросов
2. **Тип вопросов**: только открытые вопросы, требующие анализа текста
3. **Сложность**: Включи вопросы трех уровней:
   - Базовые (фактологические, 40%)
   - Средние (понимание взаимосвязей, 40%)
   - Сложные (анализ и синтез, 20%)
4. **Ответы**: Полные предложения, содержащиеся в тексте дословно или в близкой формулировке
5. **Формат**: Четкое структурирование по темам с нумерацией
Пример выполнения:
Тема: [Название темы]
1. Вопрос: [формулировка] | Уровень: [базовый/средний/сложный]
   Ответ: [полный ответ]
```

Для градации вопросов по сложности применяются следующие модификаторы промпта:

1) базовый уровень: «Сформулируй вопрос, проверяющий знание определений и основных понятий»;

2) средний уровень: «Создай вопрос, требующий объяснения причинно-следственных связей»;

3) сложный уровень: «Разработай вопрос, требующий проведение анализа преимуществ/недостатков или сравнение концепций».

Процесс валидации качества пары «Вопрос — ответ» включает проверку на соответствие следующим требованиям: 1) наличие точного ответа в исходном тексте; 2) соответствие уровня сложности заявленному; 3) отсутствие дублирования формулировок; 4) соблюдение требуемого формата вывода.

Адаптивное тестирование

На UML-диаграмме последовательности, представленной на рис. 4, показан процесс адаптивного тестирования пользователя.

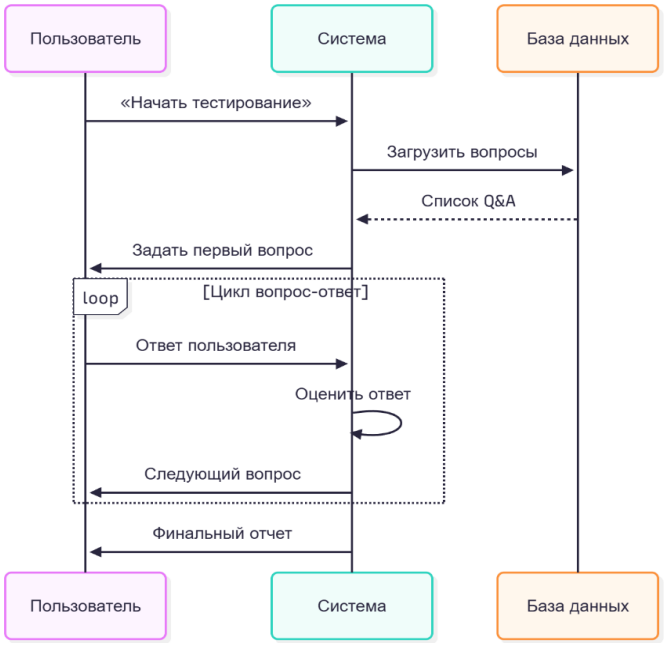

Рис. 4. UML-диаграмма последовательности: процесс тестирования пользователя

Инициализация. При получении запроса «Начать тестирование» система запускает алгоритм тестирования, который включает: 1) загрузку структурированного списка вопросов из тестовой базы данных; 2) инициализацию параметров адаптации тестирования с указанием начального уровня сложности (задается в начале тестирования), целевых показателей обучения и ограничения по количеству вопросов.

Адаптивная логика выбора вопроса. Система задает пользователю первый вопрос, после чего анализирует ответ на корректность и семантическую схожесть с эталонным ответом. Сравнение с эталонным ответом основано на базовом промпте:

Листинг 2. Пример промпта для оценки текущего ответа на вопрос

```
СИСТЕМА: Ты — модуль семантической валидации. Оцени степень соответствия ответа
пользователя эталону.
КОНТЕКСТ:
- ЭТАЛОН: "{правильный_ответ}"
- ОТВЕТ: "{ответ_пользователя}"
ШКАЛА ОЦЕНКИ:
TRUE — полное семантическое соответствие, все ключевые элементы присутствуют
PARTIAL — частичное соответствие, основные понятия сохранены, но есть неточности
FALSE — существенные расхождения или отсутствие ключевых элементов
ПРАВИЛА:
- Допускаются синонимы и перефразирования
- Учитывай контекстную уместность
- Игнорируй стилистические различия
ВЫВОД: Только одно слово — TRUE/PARTIAL/FALSE
```

Динамический подбор следующего вопроса. Далее применяется специальный алгоритм подбора вопросов, с помощью которого выполняется построение индивидуальной траектории тестирования. Общая схема алгоритма динамического подбора вопроса показана на рис 5. Для его работы применяются специализированные правила. Если пользователь дал правильный ответ, то система осуществляет переход к следующей теме тестовой базы, выбранной случайно. Если пользователь дал частично правильный ответ, то ему предлагается дополнительный вопрос аналогичного уровня

сложности, для закрепления материала. В случае, если пользователь дал неправильный ответ, то система подбирает более простой вопрос по этой тематике и предлагает его пользователю.

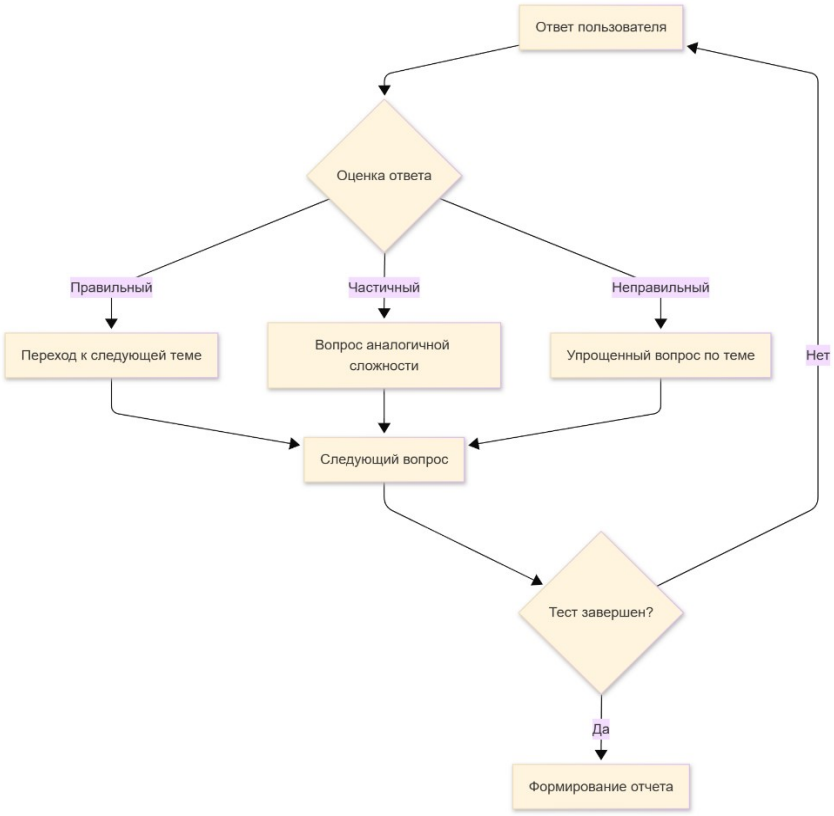

Рис. 5. Схема подбора следующего вопроса

Формирование отчета. После проведения тестирования система проводит анализ статистики по всем ответам пользователя и генерирует персонифицированные рекомендации по «проблемным» темам. Процесс анализа статистики происходит на основе базового промпта:

Листинг 3. Пример базового промпта для формирования итогового отчета

```
Проанализируй результаты тестирования и сформируй итоговый отчет:
КОНТЕКСТ:
- Вопросы и эталонные ответы: {question_answer_pairs}
- Ответы пользователя: {user_answers}
ИНСТРУКЦИИ:
1. Рассчитай общий балл (1 балл за каждый правильный ответ)
2. Определи темы с ошибками пользователя
3. Сформулируй конкретные рекомендации по повторению материала
ФОРМАТ ОТЧЕТА:
ИТОГОВЫЙ БАЛЛ: [X] из [Y] ([Z]%)
АНАЛИЗ ОШИБОК:
[Номер вопроса]. [Тема вопроса]
- Ваш ответ: [ответ пользователя]
- Правильный ответ: [эталонный ответ]
РЕКОМЕНДАЦИИ:
Для улучшения результатов рекомендуется повторить:
- [Тема 1] (ошибки в вопросах: [номера вопросов])
- [Тема 2] (ошибки в вопросах: [номера вопросов])
```

Во время анализа результатов (сравнения ответов пользователя с эталонными формулировками) система производит подсчет общего количества верных/неверных ответов, их процентного соотношения, а также группирует ошибки по тематическим разделам. Затем на основе анализа частоты ошибок формируются рекомендации путем составления перечня тем для повторного изучения, а также составляются рекомендации в привязке к конкретным разделам учебного материала. Пример отчета представлен на рис. 6.

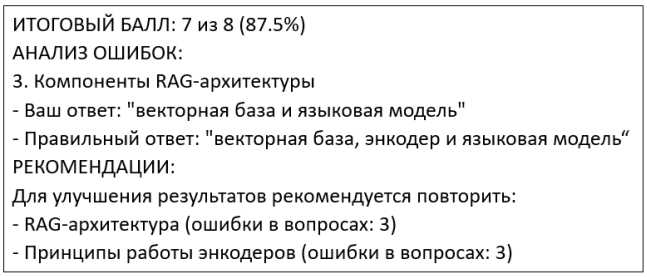

Рис. 6. Пример итогового отчета по прохождению теста

Эксперименты и обсуждение результатов

В ходе работы была реализована интеллектуальная система поддержки исследовательской деятельности. Для оценки эффективности разработанной системы был проведен ряд экспериментов, направленных на проверку функционала по автоматической генерации тестов и адаптивному тестированию.

Оценка модуля автоматической генерации тестов и адаптивного тестирования

Для проверки качества системы по автоматической генерации тестовых вопросов и адаптивному тестированию были проведены тесты на основе учебных материалов по тематике «Задачи удовлетворения ограничений» (CSP) [21].

Качество работы системы оценивалось по следующим аспектам: 1) качество генерации: релевантность, грамматическая корректность и фактическая точность сгенерированных вопросов; 2) корректность работы адаптивного алгоритма: способность системы динамически менять последовательность вопросов на основе проверки ответов.

Для оценки модуля был сгенерирован список вопросов, часть которого представлена в таблице.

На рисунке 7 представлен пример, при котором ответ пользователя система засчитывает как полностью верный. В этом случае происходит переход к другой теме вопросов.

На рисунке 8 продемонстрирован случай, когда пользователь дал частично правильный ответ. Ответ содержит назначение эвристики, но не содержит описание преимуществ от ее применения, заключающееся в минимизации количества исключаемых вариантов для соседних переменных. Система предлагает еще вопрос из той же темы. Если пользователь дал на него правильный ответ, то происходит переход к другой теме тестирования.

Если пользователь дает кардинально неверный ответ, то система подбирает вопрос из той же темы, но более низкой сложности. Далее, если ответ на вопрос правильный, то происходит переход к следующей теме с ранее установленной сложностью. Такой случай показан на рис. 9.

Демонстрация генерации отчета показана на рис. 10. На нем рассмотрен результат прохождения теста по теме «Задачи удовлетворения ограничений», состоящего из 8 вопросов.

В процессе исследований выявлено, что, что модули генерации и подбора вопросов демонстрируют высокую эффективность при проведении тестирования системы. Система корректно анализирует ответы и изменяет последовательность вопросов в соответствии с заданной логикой адаптации. Тем не менее в ряде случаев (7–9 %) у системы наблюдаются ошибки при семантической оценке ответов пользователя и арифметические ошибки при подсчете итогового балла за тест, что говорит о необходимости модернизации механизма подсчета баллов.

Фрагмент из списка сгенерированных вопросов

Тема: Основные понятия CSP		
Вопрос	Уровень	Ответ
Как определяются переменные и ограничения в задаче удовлетворения ограничений?	Базовый	Задача удовлетворения ограничений определена множеством переменных $x_1, x_2, \ldots, x_n$ и множеством ограничений $c_1, c_2, \ldots, c_n$. Каждая переменная x_i имеет непустую область определения D_i возможных значений. Каждое ограничение c_i включает некоторое подмножество переменных и задает допустимые комбинации значений для этого подмножества
Что такое «полное присваивание» и чем оно отличается от «решения» в CSP?	Средний	Полным называется такое присваивание, в котором участвует каждая переменная, а решением задачи CSP является полное присваивание, которое удовлетворяет всем ограничениям
…	…	…
Тема: Алгоритмы поиска для CSP		
Какие два основных подхода к поиску решений в CSP описаны в тексте?	Базовый	Для решения задач CSP широко применяются алгоритмы поиска в глубину (поиск с возвратами), а также могут использоваться методы локального поиска на основе формулировки с полным состоянием
…	…	…
Почему алгоритм с минимальными конфликтами эффективен для решения задачи с N ферзями?	Сложный	Алгоритм с минимальными конфликтами показал себя чрезвычайно эффективным при решении многих задач CSP, особенно при наличии подходящего начального состояния. Больше всего достойно удивления то, что при решении задачи с n ферзями время прогона алгоритма с минимальными конфликтами остается почти независимым от размера задачи. Данный алгоритм решает в среднем за 50 этапов (после начального присваивания) даже задачу с миллионом ферзей

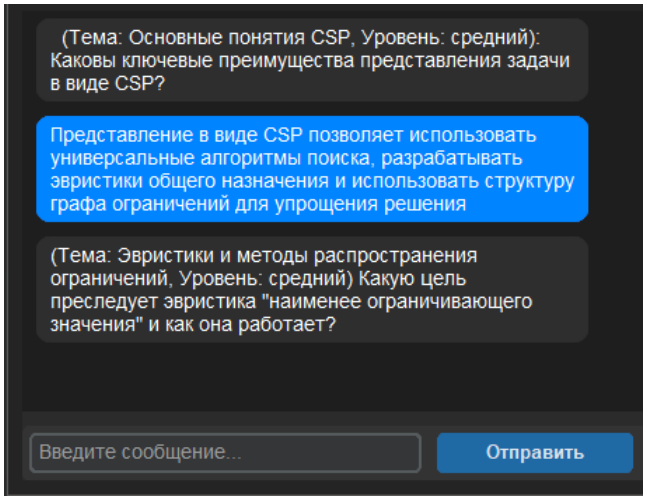

Рис. 7. Пример работы системы при правильном ответе пользователя

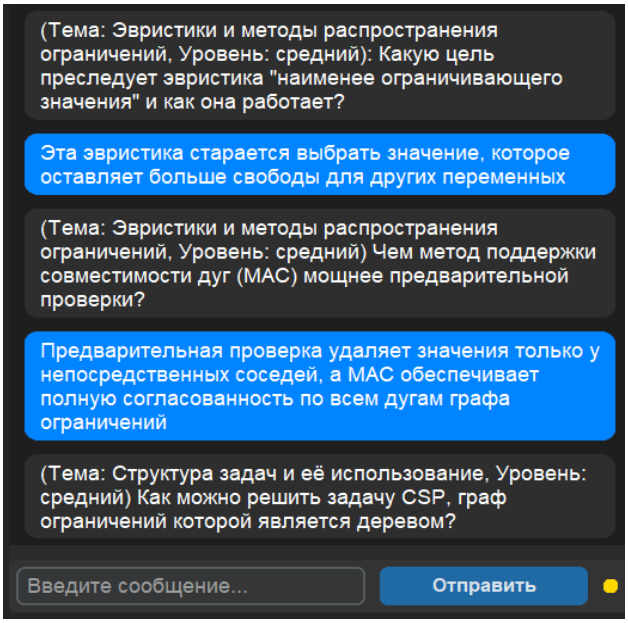

Рис. 8. Пример работы системы при частично правильном ответе пользователя

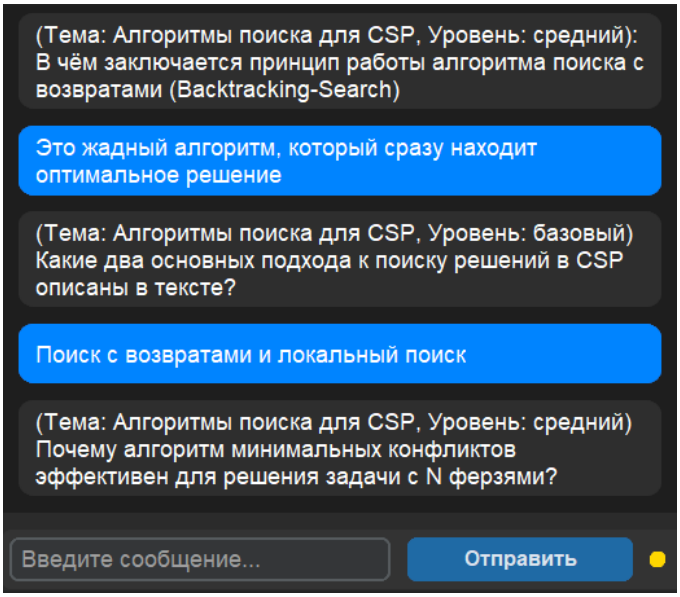

Рис. 9. Пример работы системы при неверном ответе пользователя

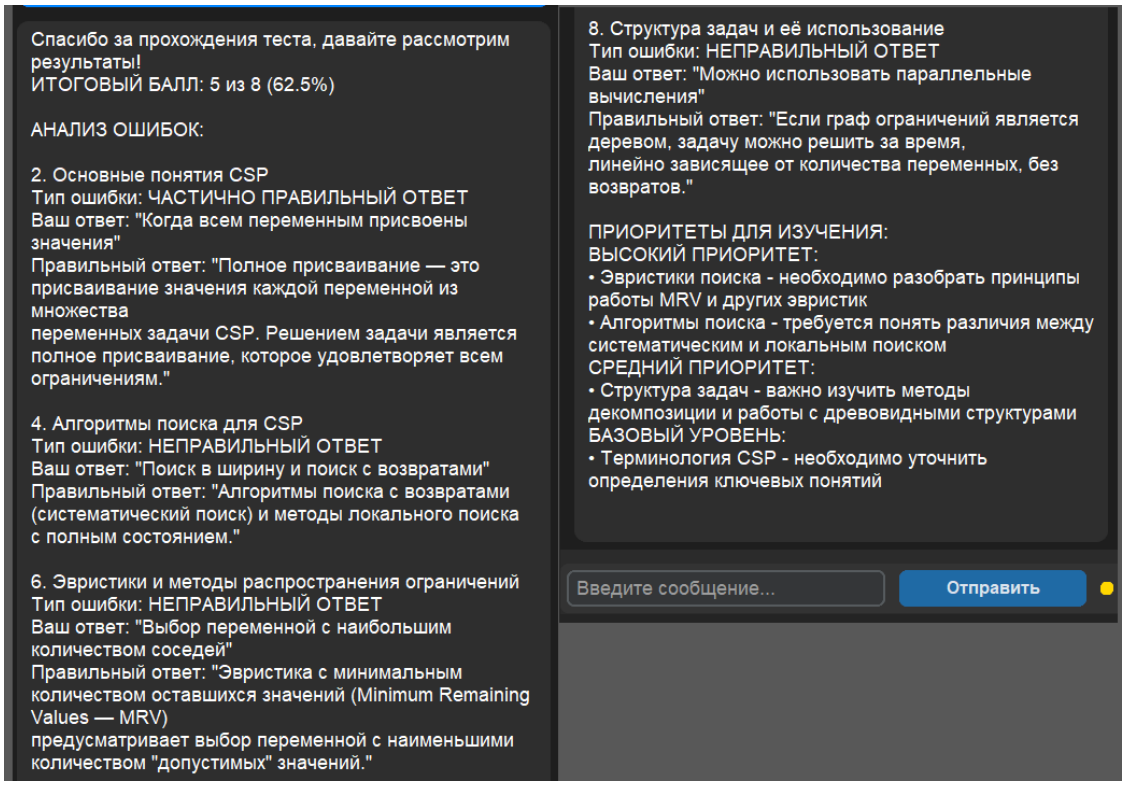

Рис. 10. Пример отчета по результатам прохождения теста

Заключение

В работе представлена система для решения задач автоматизированной генерации тестов на основе указанных пользователем документов и адаптивного тестирования. Подробно рассмотрена ее архитектура, основным компонентом которой является блок контекстно-ориентированной обработки информации. Разработанная программная система зарегистрирована в Роспатенте [22].

В работе приводятся базовые промпты для генерации тестовых вопросов, эталонных ответов, а также для проверки результатов тестирования и формирования отчетов на основании этих результатов. Исследования показали, что уже на текущем этапе система вполне удовлетворительно способна решать большинство стоящих перед ней задач, в частности регулировать сложность вопросов относительно ответов пользователя при формировании индивидуальной траектории обучения и определять уровень квалификации пользователей, что востребовано в исследовательской деятельности. Однако более интеллектуальные механизмы генерации вопросов, а также их адаптивного подбора во время тестирования, по всей видимости, требует привлечения дополнительных инструментов и технологий. На текущий момент учет и анализ контекста выполняются преимущественно на основе RAG-технологии. В дальнейшем для повышения точности обработки информации планируется применять онтологии предметных областей. В качестве еще одного направления исследований можно отметить целесообразность доработки блока арифметического подсчета результатов, улучшения модуля динамического выбора вопросов, а также реализации алгоритмов для работы с формулами.

Список источников

1. Большая языковая модель [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/%D0%91%D0%BE%D0%BB%D1%8C%D1%88%D0%B0%D1%8F_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C (дата обращения: 01.10.2025).
2. Gao Y. et al. Retrieval-Augmented Generation for Large Language Models: A Survey // arXiv:2312.10997 [cs]. 2023.
3. Elicit: The AI Research Assistant [Электронный ресурс]. URL: https://support.elicit.com/en/categories/146369 (дата обращения: 01.04.2025).
4. Semantic Scholar API Tutorial [Электронный ресурс]. URL: https://www.semanticscholar.org/product/api/tutorial (дата обращения: 12.09.2025).
5. IBM Watson Discovery [Электронный ресурс]. URL: https://www.ibm.com/products/watson-discovery (дата обращения: 03.09.2025).
6. A Retrieval-Augmented Generation Framework for Academic Literature Navigation in Data Science // arXiv:2412.15404 [cs]. 2024.
7. LitLLM: A Toolkit for Scientific Literature Review // arXiv:2402.01788 [cs]. 2024.
8. OpenAlex: Technical Documentation [Электронный ресурс]. URL: https://docs.openalex.org/ (дата обращения: 10.01.2025).
9. Learning to Ask: Neural Question Generation for Reading Comprehension // arXiv:1705.00106 [cs]. 2017.
10. LSTM-сети долгой краткосрочной памяти [Электронный ресурс]. URL: https://habr.com/ru/companies/wunderfund/articles/331310/ (дата обращения: 10.01.2025).
11. Rajpurkar P. et al. SQuAD: 100,000+ Questions for Machine Comprehension of Text // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2016. P. 2383–2392.
12. Nguyen T. et al. MS MARCO: A Human Generated Machine Reading Comprehension Dataset // arXiv:1611.09268 [cs]. 2016.
13. Pennington J. et al. GloVe: Global Vectors for Word Representation // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014. P. 1532–1543.
14. Mysore S. et al. AClimb: Annotating and Analyzing Clarification in Conversational Search // arXiv:2406.03397 [cs]. 2024.
15. ROUGE (metric) [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/ROUGE_(metric) (дата обращения: 10.01.2025).
16. Олейник А. Г. и др. Об использовании RAG-технологии для исследовательского поиска в справочных и нормативных текстах // Труды Кольского научного центра РАН. Серия: Технические науки. 2024. Т. 15, № 3. С. 5–26.
17. Faiss: A Library for Efficient Similarity Search [Электронный ресурс]. URL: https://faiss.ai/ (дата обращения: 01.11.2024).
18. Yandex Foundation Models [Электронный ресурс]. URL: https://yandex.cloud/ru/docs/foundation-models/ (дата обращения: 20.03.2025).
19. GigaChat API [Электронный ресурс]. URL: https://developers.sber.ru/docs/ru/gigachat/api/overview (дата обращения: 20.07.2025).
20. Langchain Introduction [Электронный ресурс]. URL: https://python.langchain.com/docs/introduction/ (дата обращения: 01.11.2024).
21. Рассел С., Норвиг П. Искусственный интеллект: современный подход. М.: Вильямс, 2021. 1416 с.

22. Свидетельство о государственной регистрации программы для ЭВМ № 2025682172 Российская Федерация. Программа автоматической генерации тестов и проверки знаний в научно-исследовательской деятельности / А. А. Зуенко, А. В. Шестаков; заявл. 31.07.2025; опубл. 21.08.2025.

References

1. Bol'shaya yazykovaya model' [Large language model]. (In Russ.). Available at: https://ru.wikipedia.org/wiki/%D0%91%D0%BE%D0%BB%D1%8C%D1%88%D0%B0%D1%8F_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C (accessed 01.10.2025).
2. Gao Y. et al. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs], 2023.
3. Elicit: The AI Research Assistant. Available at: https://support.elicit.com/en/categories/146369 (accessed 01.04.2025).
4. Semantic Scholar API Tutorial. Available at: https://www.semanticscholar.org/product/api/tutorial (accessed 12.09.2025).
5. IBM Watson Discovery. Available at: https://www.ibm.com/products/watson-discovery (accessed 03.09.2025).
6. A Retrieval-Augmented Generation Framework for Academic Literature Navigation in Data Science. arXiv:2412.15404 [cs], 2024.
7. LitLLM: A Toolkit for Scientific Literature Review. arXiv:2402.01788 [cs], 2024.
8. OpenAlex: Technical Documentation. Available at: https://docs.openalex.org/ (accessed 10.01.2025).
9. Learning to Ask: Neural Question Generation for Reading Comprehension. arXiv:1705.00106 [cs], 2017.
10. LSTM-seti dolgoy kratkosrochnoy pamyati [LSTM-Long Short-Term Memory networks]. (In Russ.). Available at: https://habr.com/ru/companies/wunderfund/articles/331310/ (accessed 10.01.2025).
11. Rajpurkar P. et al. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 2383–2392.
12. Nguyen T. et al. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. arXiv:1611.09268 [cs], 2016.
13. Pennington J. et al. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.
14. Mysore S. et al. AClimb: Annotating and Analyzing Clarification in Conversational Search. arXiv:2406.03397 [cs], 2024.
15. ROUGE (metric). Available at: https://en.wikipedia.org/wiki/ROUGE_(metric) (accessed 10.01.2025).
16. Oleynik A. G. et al. Ob ispol'zovanii RAG-tekhnologii dlya issledovatel'skogo poiska v spravochnyh i normativnyh tekstah [On the use of RAG technology for research search in reference and regulatory texts]. *Trudy Kol'skogo nauchnogo centra RAN. Seriya: Tehnicheskie nauki* [*Proceedings of the Kola Science Centre of the Russian Academy of Sciences. Series: Engineering Sciences*], 2024, vol. 15, no 3, pp. 5–26. (In Russ.).
17. Faiss: A Library for Efficient Similarity Search. Available at: https://faiss.ai/ (accessed 01.11.2024).
18. Yandex Foundation Models. Available at: https://yandex.cloud/ru/docs/foundation-models/ (accessed 20.03.2025).
19. GigaChat API. Available at: https://developers.sber.ru/docs/ru/gigachat/api/overview (accessed 20.07.2025).
20. Langchain Introduction. Available at: https://python.langchain.com/docs/introduction/ (accessed 01.11.2024).
21. Russell S., Norvig P. *Iskusstvennyy intellekt: sovremennyy podkhod* [Artificial Intelligence: A Modern Approach]. Moscow, Williams Publ., 2021, 1416 p. (In Russ.).
22. Certificate of state registration of the computer program No. 2025682172 Russian Federation. *Programma avtomaticheskoy generatsii testov i proverki znaniy v nauchno-issledovatel'skoy deyatel'nosti* [Program for automatic test generation and knowledge testing in research activities]. A. A. Zuenko, A. V. Shestakov, declared 31.07.2025, published 21.08.2025.

Информация об авторах

А. В. Шестаков — аспирант, стажер-исследователь;
А. А. Зуенко — кандидат технических наук, ведущий научный сотрудник.

Information about the authors

A. V. Shestakov — Graduate Student, Intern Researcher;
A. A. Zuenko — Candidate of Science (Tech.), Leading Scientific Officer.

Статья поступила в редакцию 01.11.2025; одобрена после рецензирования 24.11.2025; принята к публикации 28.11.2025.
The article was submitted 01.11.2025; approved after reviewing 24.11.2025; accepted for publication 28.11.2025.